



Rozšiřitelnost ASP.NET MVC

Tomáš Pastorek, @topasCZ

billy
LAS VEGAS

alexandro
SÃO PAULO

أحمد
DUBAI

elroy
CAPE TOWN

clément
PARIS

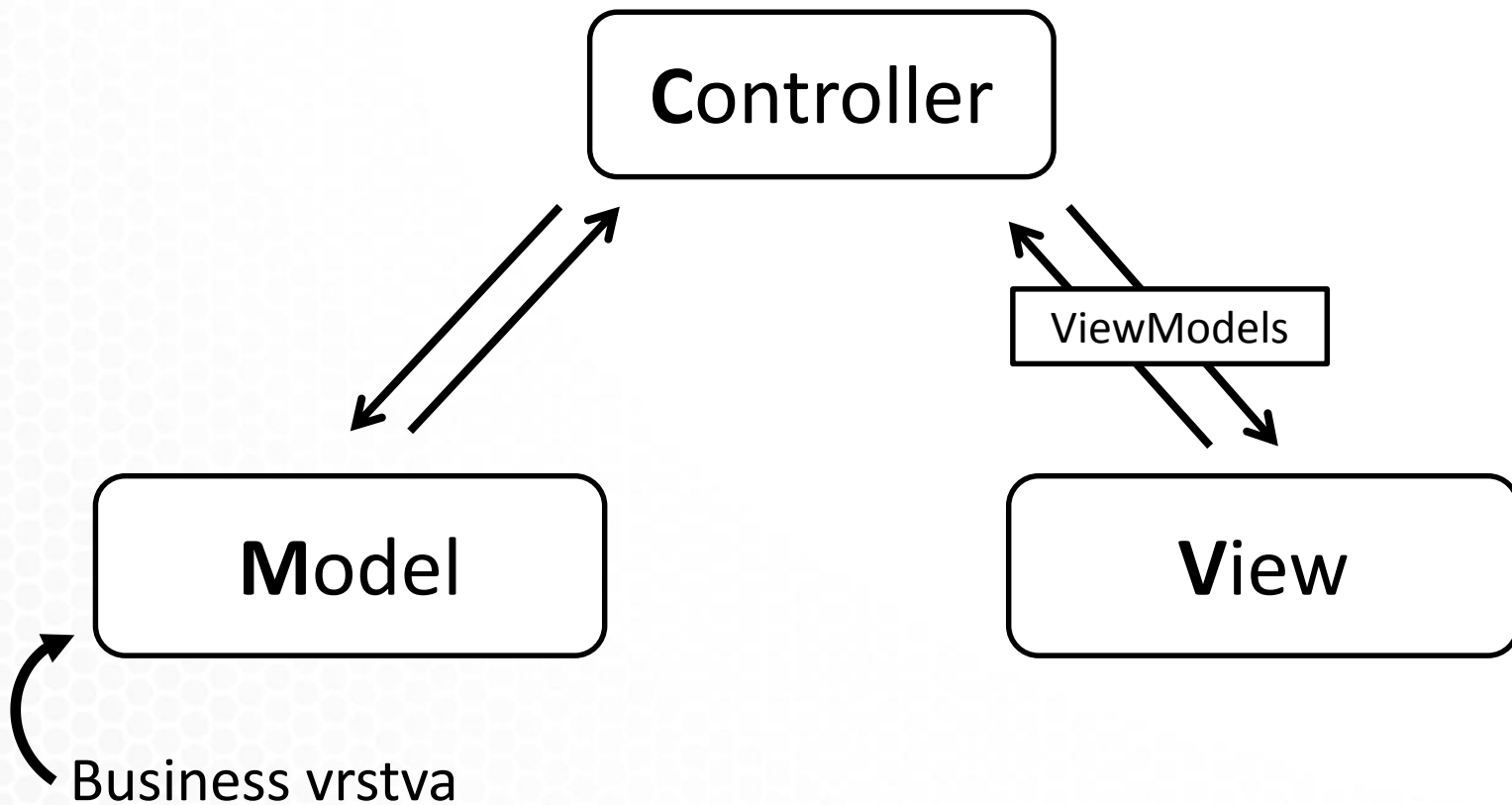
幸洋
TOKYO

amanda toh
SINGAPORE

denisa
PRAGUE

梁亞麗
HONG KONG

ASP.NET MVC



Lifecycle ASP.NET MVC

1. URL request (GET, POST, ...)
2. Routování (System.Web.Routing):
 1. RouteTable (RouteCollection)
 - Seznam routovacích pravidel (třída Route)
 2. MvcRouteHandler : IRouteHandler
 - Metoda pro získání IHttpHandleru
 3. MvcHandler : IHttpHandler
 - Pomocí IControllerFactory vytvoří instanci controller a spustí...



Lifecycle ASP.NET MVC

3. Controller.Execute(RouteContext ctx)

- Zpracování akční metody (IActionInvoker)
 1. Výběr akční metody
 2. Spouštění filtrů
 3. ModelBinding
 4. Vrácení výsledku *)

*) ViewResult používá kolekci IViewEngine pro render stránky

Routování

Namespace System.Web.Routing

- **Route** – Oboustranný převod mezi URL a datovými strukturami routování.
- **IRouteConstraint** – Omezení nad routovacím pravidlem (vrací 404 v případě chyby)
- **IRouteHandler** – Vlastní handler pro obsluhu požadavku.

Projekt RouteMagic (Phil Haack)

<https://github.com/haacked/routemagic>

RouteDebugger (NuGet)



Demo

Vytváření controllerů

- Implementovat **IControllerFactory**
- Lépe zdědit z **DefaultControllerFactory**
 - přepsat metody `GetControllerInstance` a `ReleaseController`
- Nastavit factory:
`ControllerBuilder.Current.SetControllerFactory(...)`
- **IDependencyResolver**
 - adapter mezi DI kontejnery a ASP.NET MVC
 - `DependencyResolver.SetResolver(...)`
 - Nutno registrovat i ostatní závislosti pro ASP.NET MVC
 - Hotové balíčky pro (téměř) každý DI kontejner



Demo

ActionFilters

- Attribute implementující jeden z těchto interface:
 - **IAuthorizationFilter** (AuthorizeAttribute, ValidateAntiForgeryTokenAttribute)
 - **IExceptionHandler** (HandleErrorAttribute, OutputCacheAttribute)
 - **IResultFilter** (OutputCacheAttribute)
 - **IActionFilter** (OutputCacheAttribute)

Použití ActionFilters

- Zdědit z **FilterAttribute**
 - Aplikace na celý controller nebo akční metodě
- Přidat do kolekce **GlobalFilters.Filters**
- Implementace **IFilterProvider**
 - Metoda `GetFilters(...)` vrací filtry, které mají být použity
 - Přidat do kolekce **FilterProviders.Providers**



Demo

Výběr akční metody

- **HttpGetAttribute, HttpPostAttribute**
(HttpPut, HttpDelete, ...)
- **ActionMethodSelectorAttribute**
 - `bool IsValidForRequest`
(ControllerContext controllerContext,
MethodInfo methodInfo)

Demo

ModelBinders

- Převod dat z requestu (POST, GET, ...) do akční metody
- **DefaultModelBinder**
 - mapování jednoduchých i složených modelů
- Implementuje **IModelBinder**
- `object BindModel(ControllerContext controllerContext, ModelBindingContext bindingContext)`

Použití ModelBinders

- Přidání do kolekce **ModelBinders.Binders**
 - Dvojice **Type**, **IModelBinder**
- Implementace **IModelBinderProvider**
 - **IModelBinder** **GetBinder(Type modelType)**
 - Přidat do kolekce **ModelBinderProviders.BinderProviders**



Demo

ActionResults

- **ViewResult, RedirectToRouteResult, RedirectToRouteResult...**

Přidáno v ASP.NET MVC 3:

- **HttpNotFoundResult, HttpStatusCodeResult, HttpUnauthorizedResult**

Uspadnění vracení hodnot z akčních metod

- **JsonResult, ContentResult, FileContentResult, FileStreamResult**
- Vlastní implementaci zdědit z **ActionResult**
 - Přepsat metodu `ExecuteResult(ControllerContext context)`

Demo

ViewEngine

- Vestavěný **RazorViewEngine** a **WebFormsViewEngine**
- Zodpovědný za vytvoření instance **IView**
- Seznamy cest pro vyhledávání souborů View:
 - `ViewLocationFormats` ("~/Views/{1}/{0}.cshtml")
 - `PartialViewLocationFormats` ("~/Views/{1}/{0}.cshtml")
 - `MasterLocationFormats` ("~/Views/Shared/{0}.cshtml")
- Kolekce enginů, které se používají: **ViewEngines.Engines**



Demo

DisplayModeProvider

- Používá vestavěný **VirtualPathProviderViewEngine**
- Mobilní web snadno a rychle: `Index.Mobile.cshtml`
- **DisplayModeProvider.Instance.Modes**
 - Kolekce pravidel – použije se první nalezené pravidlo
- Pravidla zděděné z **IDisplayMode**
- Jednoduchá implementace pomocí **DefaultDisplayMode**
 - Název módu
 - Lambda pro určení, kdy se má mód použít

Browser overriding

- Extenzní metody pro **HttpContextBase**
 - `SetOverridenBrowser(BrowserOverrider browser)`
 - `ClearOverridenBrowser()`
 - A další...
- Namespace `System.Web.WebPages`
- Přepisují nastavení `UserAgent`
- Ukládají do cookie `".ASPXBrowserOverride"`



Demo

Extenzní metody

- **HtmlHelper**
 - Generování HTML kódu
 - Vrací MvcHtmlString : IHtmlString
- **UrlHelper**
 - Pomocné metody pro práci s URL a routováním
- Pro helpery z vlastních namespaces je nutné
 - @using
 - Web.config v adresáři View:
< system.web.webPages.razor><pages><namespaces>

Tvorba helperů

- Třída **TagBuilder**
- Statické metody **HtmlHelper**
 - `AnonymousObjectToHtmlAttributes`
- Namespace `System.Web.Mvc.Html`:
 - Extenzní metody pro `HtmlHelper`:
 - `Id`, `IdFor`, `Name`, `NameFor`
- Razor kód jako parametr metody (Razor delegate):
`Func<object, HelperResult>`
- **`ModelMetadata`** – práce s anotacemi a metadatami modelu

Tvorba helperů

- Blokované (kontejnerové) helpery:
Disposable + disposable pattern
- Konvence `Begin*`, `End*`
- Metoda `Begin*` je extenzní metoda `HtmlHelperu`
- Html kód se vkládá pomocí `ViewContext.Writer`

Tvorba “fluent” helperů

- Třída “TestHelper” dědí z `IHtmlString`
 - Html kód vrací metoda `ToHtmlString()`
- Settery:

```
public TestHelper Text(string text)
{
    Text = text;
    return this;
}
```
- Extenzní metoda pro `HtmlHelper`, která vrátí `TestHelper`

Vlastní ViewPageBase

- Zdědit z `WebViewPage<T>` (Razor)
- Snáze přístupné vlastní helpery nebo metody ve View
- Silně typové zabalení `ViewBag/ViewData`
- Web.config:

```
<system.web.webPages.razor><pages  
pageBaseType=“...”>
```

Demo

Editor/display templates

- `Html.EditorFor(...)`
- `Html.DisplayFor(...)`
- DRY koncept
- Vestavěné DisplayTemplates: Boolean, Decimal, EmailAddress, HiddenInput, Html, Object, string, Url
- EditorTemplates: Boolean, Decimal, HiddenInput, MultilineText, Object, Password, String

Editor/display templates

- Priority cest:
 1. Area...Views/Controller/DisplayTemplates/Template.cshtml
 2. Area...Shared/DisplayTemplates/Template.cshtml
 3. Views/Controller/DisplayTemplates/Template.cshtml
 4. Views/Shared/DisplayTemplates/Template.cshtml

Editor/display templates

- Priorita názvů souborů šablon:
 - `ModelMetadata.TemplateHint`
 - `ModelMetadata.DataTypeName`
 - Název typu: (`Type.Name`)
 - Šablona musí být připravena i pro nullable typy, `ModelMetadata.IsNullableValueType`
 - Názvy jednoduchých typů: např. “String”
 - Složené typy: “Object” (hledání v dědičnosti)



Demo

Javascriptová validace

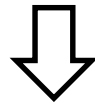
- Unobtrusive volání
- jQuery validate + wrapper (jquery.validate.unobtrusive.js)
- Validační atribut s klientskou validací:
 1. Pro poděděné atributy z výchozích anotací:
DataAnnotationsModelValidatorProvider
 2. ValidationAttribute, IClientValidatable
+ implementace GetClientValidationRules a
validačních pravidel v JS a wrapperu



Demo

Razor 2 - URL

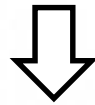
```
<script  
src="@Url.Content("~/Scripts/script.js")">  
</script>
```



```
<script src="~/Scripts/script.js"></script>
```

```

```



```

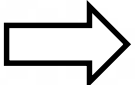
```

Razor 2 - Atributy

```
@{string trida = null;}  
<div class="@trida">...
```

```
<div class="">...            <div>...
```

```
@{ string a = null; string b = "bar" }  
<div class="@a @b">
```

```
<div class=" b">...            <div class="b">...
```

Boolean atributy: checked, readonly

Alternativní frameworky

- NancyFx (inspirace Sinatra)
 - Jednoduchý a lehký framework pro tvorbu backend
 - Hosting: ASP .NET, OWIN, Self, WCF
 - Pro ASP .NET jako `HttpHandler`
- Nina, Manos
- FubuMVC
- OpenRasta

Tipy na závěr

- Nebojte se dělat helpery – DRY
- Editor/Display templates (pouze pro opakující se případy)
- Moc chování ASP.NET MVC neupravovat, protože s tím mohou počítat některé balíčky 😊

- Mohu doporučit:
 - NuGet
 - T4MVC
 - TwitterBootstrapMvc
 - FluentValidation

Děkuji za pozornost

www.aspnetmvc.cz

@aspmvc

tomas@tomaspastorek.cz, @topasCZ